



Vidyavardhini's College of Engineering and Technology, Vasai

Department of Computer Science & Engineering (Data Science)

Experiment No. 7
Create an immersive environment (living room/ battlefield/ tennis court) with only static game objects. 3D game objects can be created using Blender or use available 3D models
Date of Performance:
Date of Submission:
Marks:
Sign:

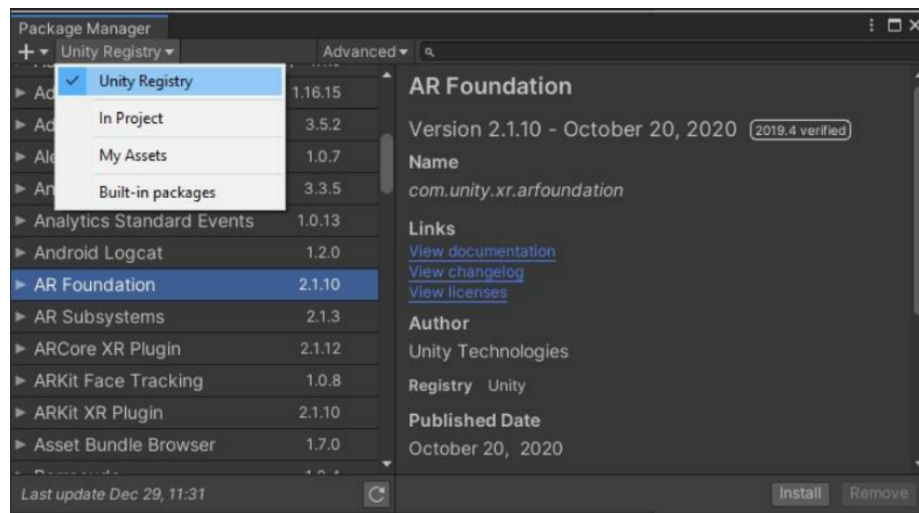


Aim: Create an immersive environment (living room/ battlefield/ tennis court) with only static game objects. 3D game objects can be created using Blender or use available 3D models

Theory:

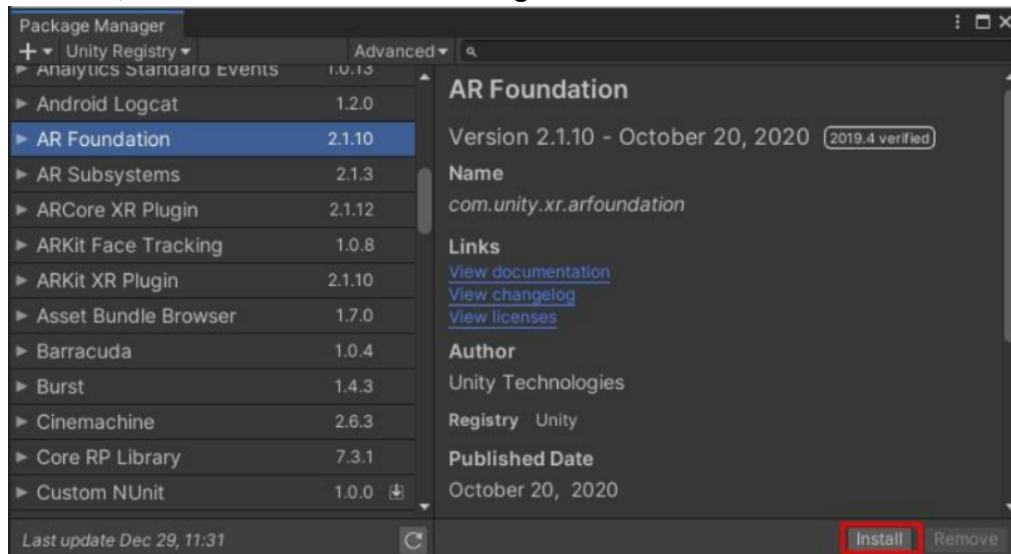
Find Packages in Unity3d

- Open unity3d create new unity project.
- Go to Window > Package Manager.
- Once the package manager window is open, ensure that you can see the package in unity registry.



Add Required Packages In Unity3D

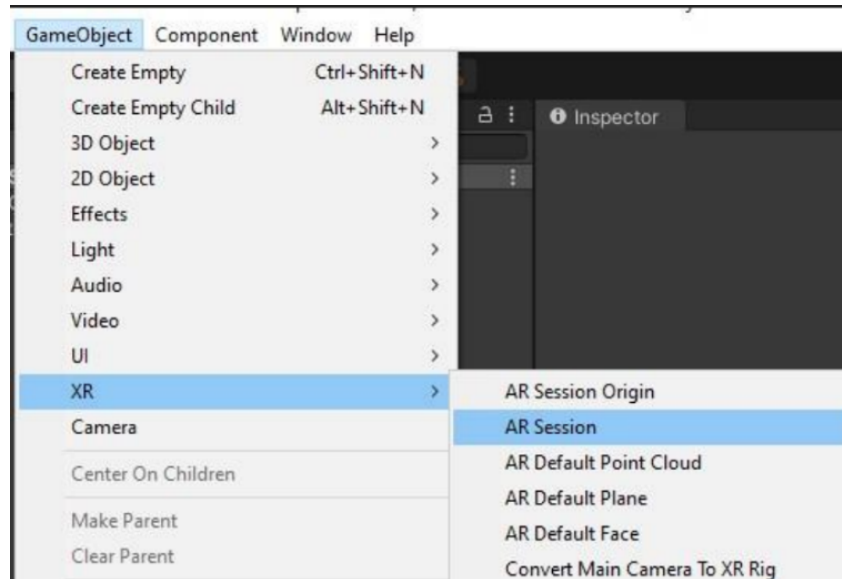
- Search for the AR Foundation and install.
- You now have the AR Foundation package in your project. For this exercise, we will be using an android device, so install the ARCore XR Plugin.



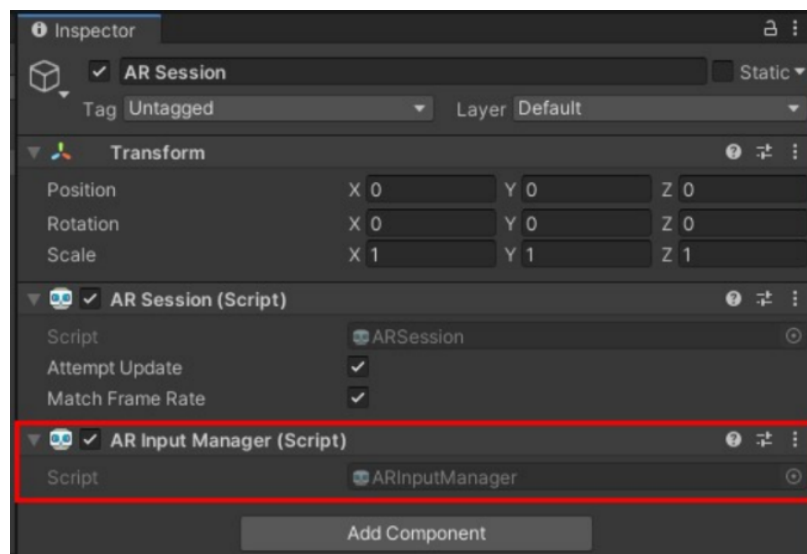


Scene Setup

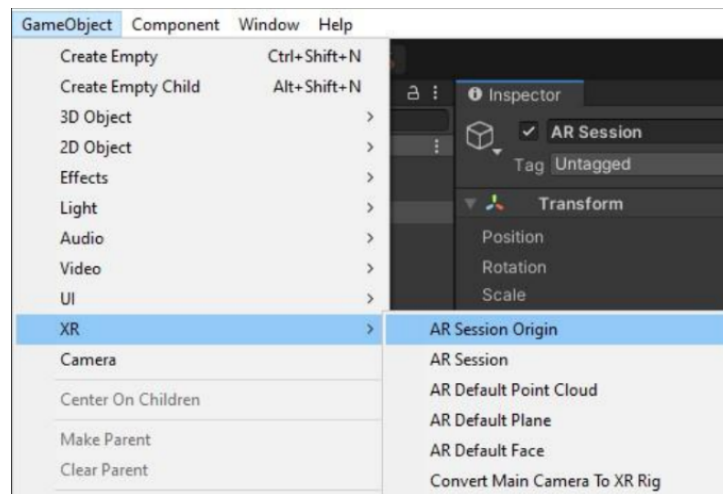
- Add the AR Session to your Scene by clicking GameObject > XR > AR Session.



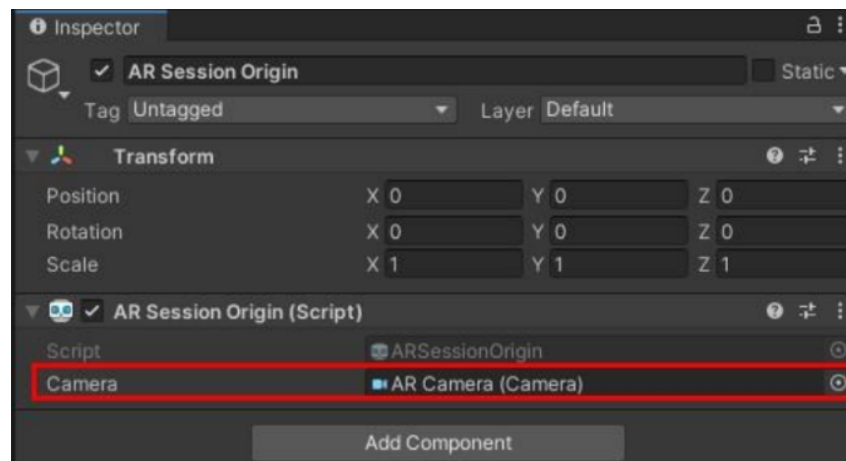
- If your AR Session does not have an AR Input Manager, Add it by clicking Add Component and searching for AR Input Manager, The AR Input Manager handles all of your scene input so you can interact with trackable objects.



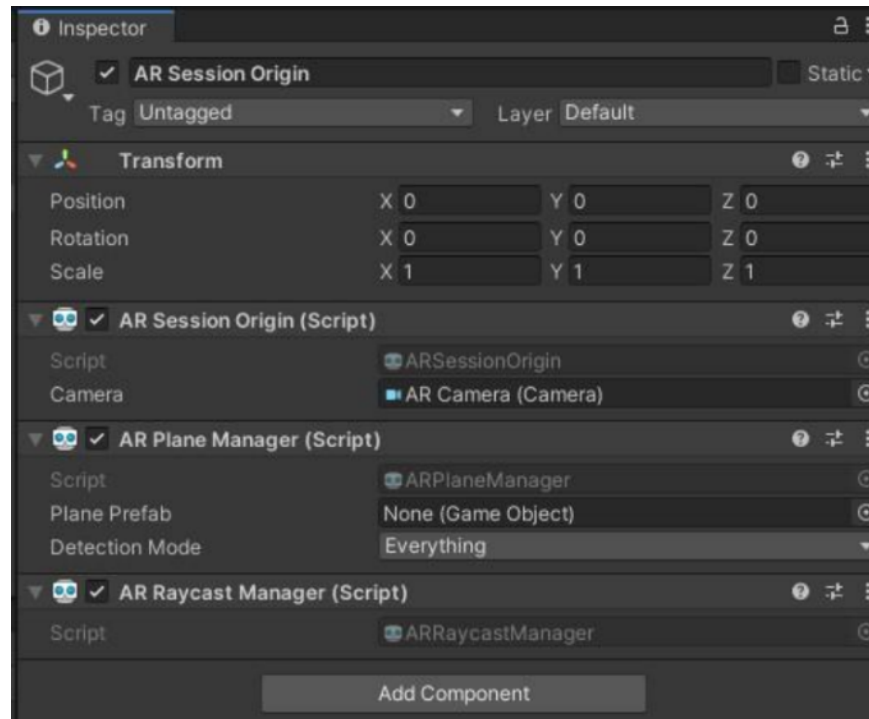
- Add the AR Session Origin Game Object to the Scene by clicking GameObject > XR > AR Session Origin



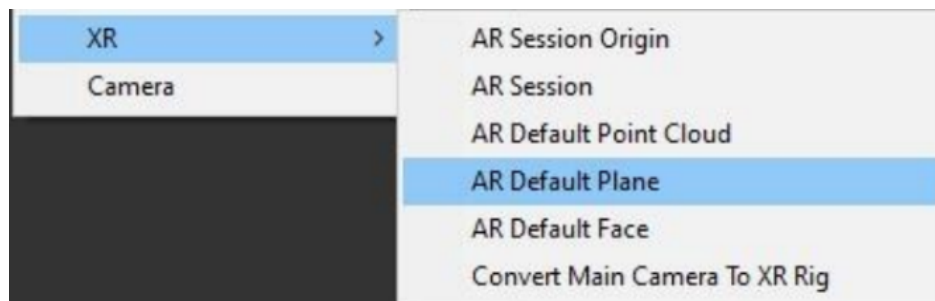
- In the AR Session Origin Inspector, make sure AR Camera is set as the reference in the AR Session Origin Camera parameter.



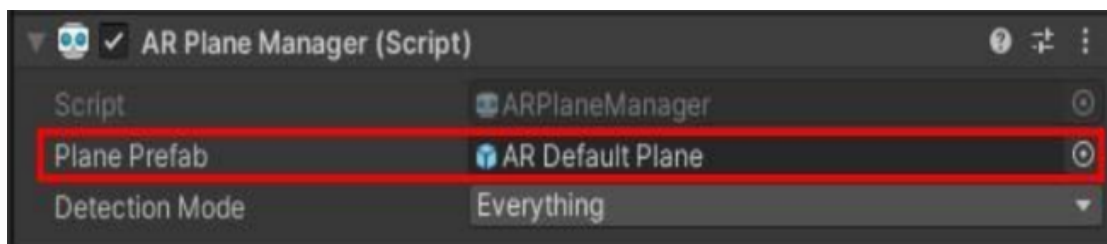
- To enable plane detection in your AR Scene, add the AR Plane Manager to the AR Session Origin object.
- Specify the AR Plane Manager to detect vertical and horizontal planes by adjusting its Detection Mode to Everything.
- To interact with trackable feature, add the AR Raycast Manager.
- Add AR Anchor Manager Component to place object on plane.



- We'll need a plane prefab so the AR Plane Manager script can display the location of the planes in our AR Scene. In the Hierarchy, right-click and select XR > AR Default Plane.

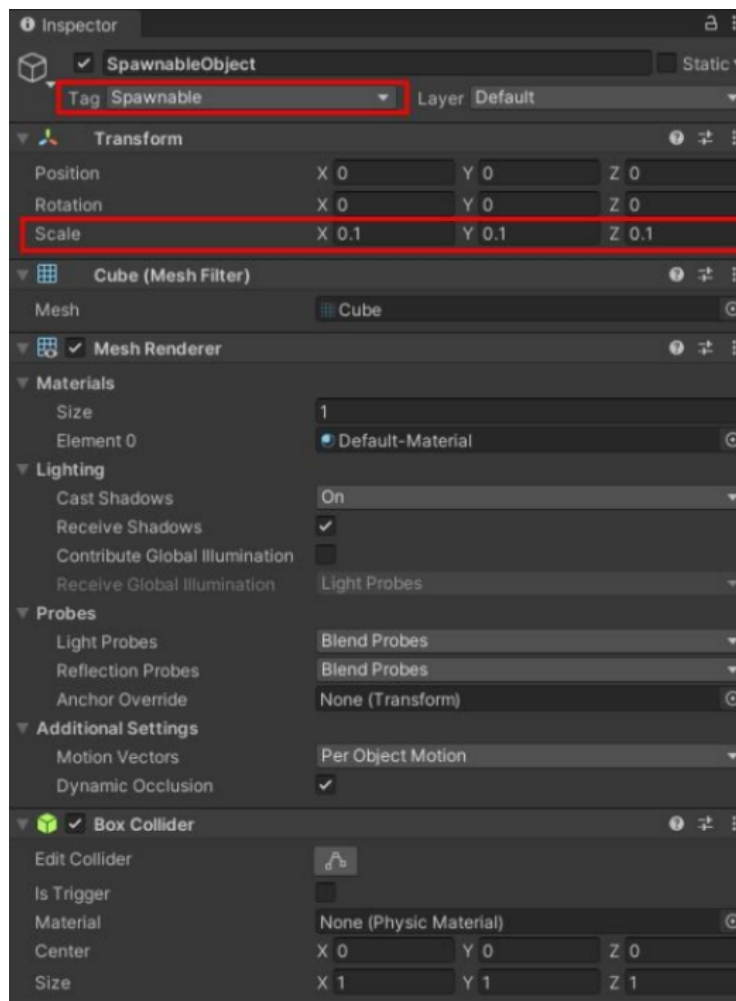


- Drag this new plane from the hierarchy into the project window to create Prefab and delete it from the hierarchy.
- In the AR Plane Manager script, assign the AR Default Plane Prefab from the project window to the Plane Prefab variable in the inspector





- We need an object to spawn in our scene. Create a Pawn and rename it Spawnable Object.
- Set the Scale to 0.1,0.1,0.1
- Create a tag name Spawnable and assign it to SpawnableObject. We will use that tag to tell when a ray hits one of these Prefabs.
- Drag SpawnableObject from the hierarchy into the Project Window to create a Prefab and then delete SpawnableObject from the Hierarchy.



Creating Script

- Create C# script and name it SpawnableManager.
- Assign the SpawnableManager script to SpawnableObject Game Object.
- Add AR Anchor Manager Component to your AR Session Origin.
- Assign SpawnableObject prefab to AR Anchor Manager's Anchor Prefab.

Writing a Script

- Adding Required NameSpaces



- using UnityEngine.XR.ARFoundation;
- using UnityEngine.XR.ARSubsystems;
- Adding Some Variables
 - ARPlaneManager m_PlaneManager; //Store AR Plane Manager component
 - ARRaycastManager m_RayCastManager; //Store AR Raycast Manager Component
 - ARAnchorManager m_ReferencePointManager; //Store AR Anchor Manager Component
 - List<ARAnchor> m_ReferencePoint; //Create List of AR Anchors.
 - static List<ARRaycastHit> s_Hits = new List<ARRaycastHit>(); // Store AR Raycast Hit

Writing a Script

- Creating Awake()

```
private void Awake()
```

```
{  
m_RayCastManager = GetComponent<ARRaycastManager>();  
m_ReferencePointManager = GetComponent<ARAnchorManager>();  
m_PlaneManager = GetComponent<ARPlaneManager>();  
m_ReferencePoint = new List<ARAnchor>();  
}
```

- Creating RemoveAllReferencePoint()

```
public void RemoveAllReferencePoint()
```

```
{  
foreach(var referencePoint in m_ReferencePoint)  
{  
//m_ReferencePointManager.RemoveAnchor(referencePoint); //Deprecated  
Destroy(referencePoint);  
}  
m_ReferencePoint.Clear();  
}
```

- Creating TryGetTouchPosition()

```
bool TryGetTouchPosition(out Vector2 touchPosition)
```

```
{  
if(Input.touchCount > 0)  
{  
touchPosition = Input.GetTouch(0).position;  
return true;  
}  
touchPosition = default;  
return false;  
}
```

- Creating Update()

```
void Update()
```

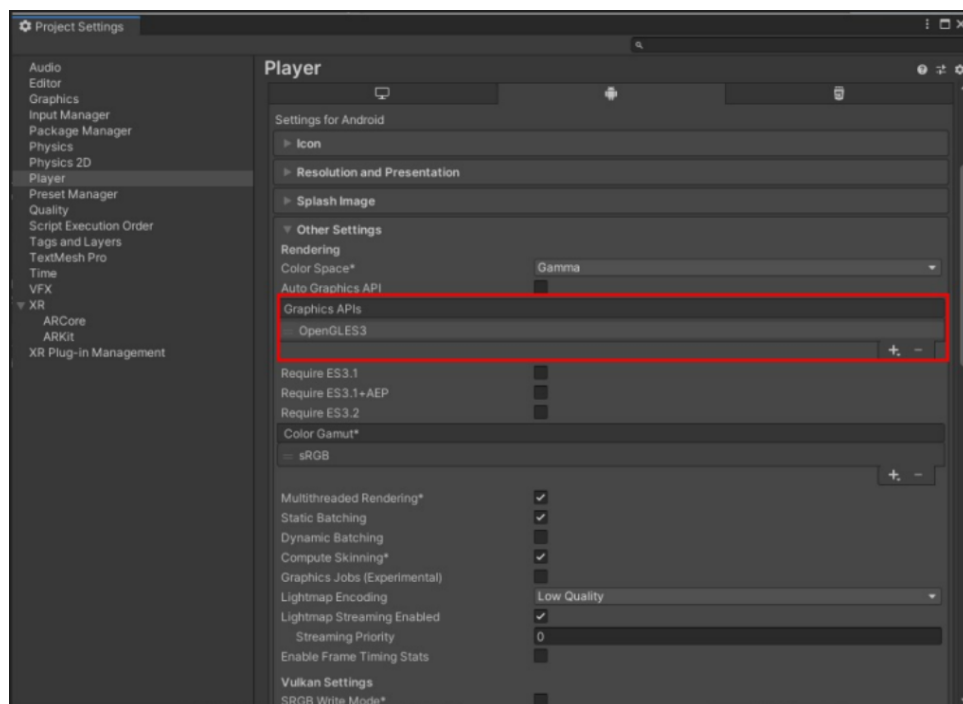
```
{  
if (!TryGetTouchPosition(out Vector2 touchPosition))  
return;  
if(m_RayCastManager.Raycast(touchPosition,s_Hits,TrackableType.Planes))
```




```
{  
var hitPose = s_Hits[0].pose;  
TrackableId planeId = s_Hits[0].trackableId; // Get the Id of Plane hit by raycast  
var  
referencePoint  
m_ReferencePointManager.AttachAnchor(m_PlaneManger.GetPlane(planeId),  
hitPose);  
if(referencePoint != null)  
{  
RemoveAllReferencePoint();  
m_ReferencePoint.Add(referencePoint);  
}  
}  
}
```

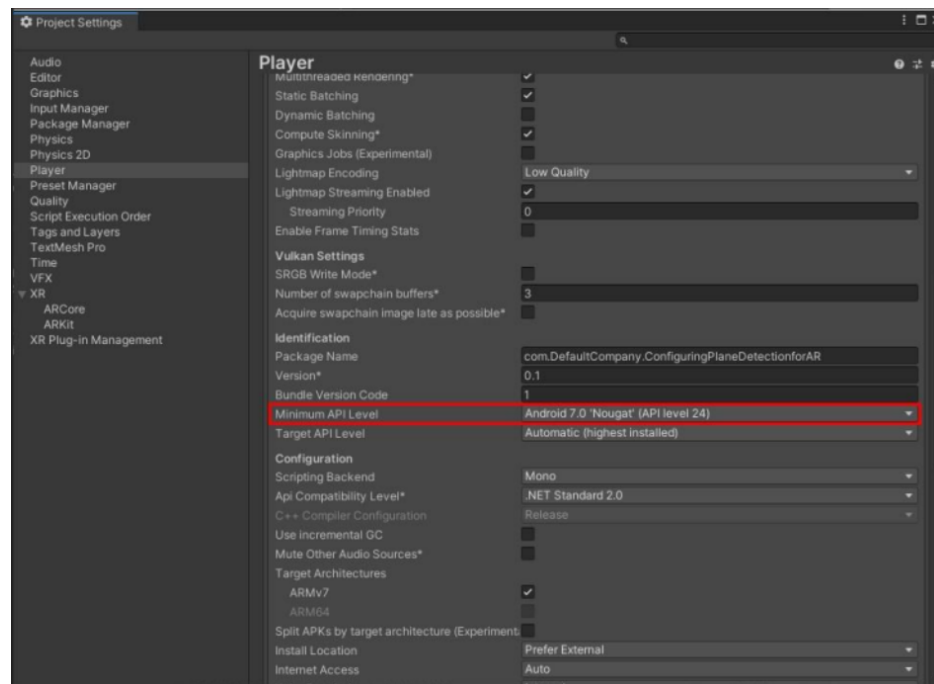
Configure Build

- Go to Edit > Project Setting > Player > Other Settings and go to graphics API Section and remove Vulkan API from the list.

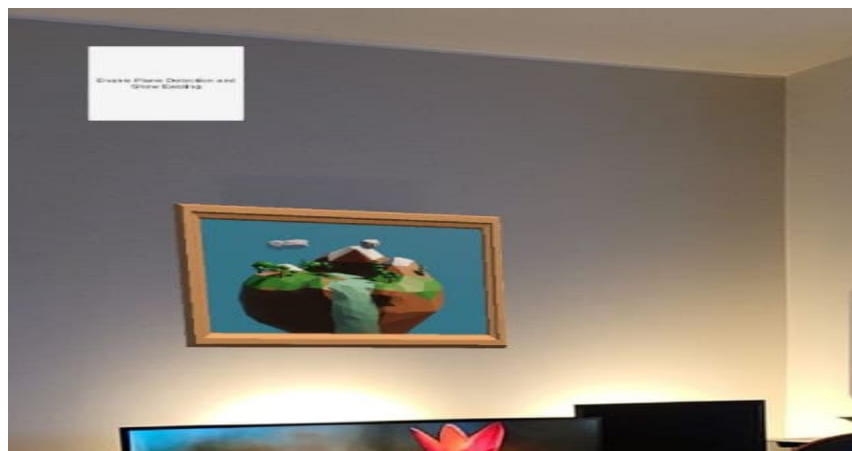


Configure Build

- Set Minimum API Level to API Level 24 or Higher.
- Close PlayerSetting Window and click Build (File->Build) using android platform.



Output:





Vidyavardhini's College of Engineering and Technology, Vasai

Department of Computer Science & Engineering (Data Science)

Conclusion:-

In Unity, we successfully crafted an immersive environment utilizing static game objects to create a vivid scene resembling a living room, battlefield, or tennis court, each meticulously designed with careful selection of 3D models. This experiment underscores the power of Unity for crafting immersive virtual environments, even with static elements. These environments lay the groundwork for interactive simulations, architectural visualizations, and gaming scenarios, demonstrating the potential of Unity as a versatile tool for creating diverse and captivating virtual worlds.